http2: overload existing http keywords to support http/2

10/12/2020 11:07 AM - Victor Julien

Meta tickets. Please create evaluate all existing http keywords and see if we can support them in http/2. For the ones we can, please create a sub-ticket each keyword. For the ones we can't support we need an explanation of why (in this ticket) and a documentation update in the user guide.

History

#1 - 11/06/2020 07:14 PM - Victor Julien
- Priority changed from Normal to High

#2 - 11/07/2020 01:25 PM - Victor Julien
- Subject changed from http/2: overload existing http keywords to support http/2 to http2: overload existing http keywords to support http/2

#3 - 12/04/2020 04:30 PM - Victor Julien
- Parent task set to #4201

#4 - 12/29/2020 08:38 PM - Philippe Antoine
- Status changed from Assigned to In Review

https://github.com/OISF/suricata/pull/5669

#5 - 03/11/2021 08:15 AM - Philippe Antoine
Some keywords are now working. Need to complete with all keywords

#6 - 04/26/2021 12:43 PM - Philippe Antoine
One keyword may not be translated:
http.stat_msg has no HTTP2 meaning (besides translating the status code)

#7 - 04/26/2021 12:46 PM - Philippe Antoine
http.start does not seem to make sense for HTTP2 in my humble opinion

#8 - 04/26/2021 01:08 PM - Philippe Antoine
http.response_body seems to be handled by file_data for HTTP2

#9 - 04/26/2021 01:10 PM - Philippe Antoine
There is no such thing as http.response_line in HTTP2

#10 - 04/26/2021 01:13 PM - Philippe Antoine
nor http.request_line
There is no http.protocol, or it is always HTTP2

#11 - 04/26/2021 01:19 PM - Philippe Antoine
http.request_body should be covered by file_data

#12 - 04/26/2021 01:38 PM - Philippe Antoine
Should HTTP2MimicHttp1Request translate headers names ? like Host becomes :authority from HTTP1 to HTTP2

How would we do the http.host normalisation ?

Should we concatenate the values in case there are multiple times the same header (name) in HTTP2 ?

#13 - 04/26/2021 01:41 PM - Philippe Antoine
http.cookie calls DetectAppLayerMpmRegister2 with SIG_FLAG_TOCLIENT and HTP_REQUEST_HEADERS, that should rather be HTP_RESPONSE_HEADERS, right ?

#14 - 05/21/2021 12:45 PM - Victor Julien
https://github.com/OISF/suricata/pull/6087 was merged towards this ticket. It is not complete as some body keywords are missing as mentioned in the PR:

- http.request_body and http.response_body, covered by file_data

#15 - 05/21/2021 01:02 PM - Philippe Antoine
What remains to be done :

- http.host : do the same normalization... same for http.header. For http.header.raw it is not raw in HTTP2, we need to concatenate key and value. For http.header_names, we can have linefeeds in HTTP2 header names, should we escape them ?
- Concatenate when we get multiple values for one header name cf https://suricata.readthedocs.io/en/suricata-6.0.0/rules/http-keywords.html#id2 example request with 2 Hosts ?
- Make HTTP2MimicHttp1Request translate header names (Host becomes :authority) ?
- http.request_body and http.response_body, covered by file_data. Should we have these specifically ?
- http.request_line and http.response_line do not exist in HTTP2, should we emulate them ? What about http.start ?
- http.protocol and http.stat_msg are implicit, should we emulate them ?

#16 - 07/14/2021 07:55 AM - Philippe Antoine
After https://github.com/OISF/suricata/pull/6183
There will be the following questions where we want the opinion of signature writers :
- http.request_body and http.response_body, covered by file_data. Should we have these specifically ?
- http.request_line and http.response_line do not exist in HTTP2, should we emulate them ? What about http.start ?
- http.protocol and http.stat_msg are implicit, should we emulate them ?

#17 - 08/27/2021 08:12 PM - Jason Williams

There will be the following questions where we want the opinion of signature writers :

- http.request_body and http.response_body, covered by file_data. Should we have these specifically ?
- http.request_line and http.response_line do not exist in HTTP2, should we emulate them ? What about http.start ?
- http.protocol and http.stat_msg are implicit, should we emulate them ?

To retain compatibility of current rules in ET ruleset for example, emulation would likely be necessary of all of these keywords are used fairly heavily

#18 - 09/06/2021 07:33 PM - Philippe Antoine
Merge of https://github.com/OISF/suricata/pull/6328

Only emulation remaining